INICIOS Y EVOLUCIÓN DE LOS PROCESADORES GRÁFICOS (GPUs)

X JORNADAS DE DIVULGACIÓN DE APLICACIONES CIENTÍFICAS Y VISIÓN POR COMPUTADOR SOBRE PROCESADORES GRÁFICOS

UNIVERSIDAD DE ALICANTE

Pablo Martínez-González <pmartinez@dtic.ua.es>

Albert García-García <agarcia@dtic.ua.es>

José García-Rodríguez <jgarcia@dtic.ua.es>

PABLO MARTÍNEZ GONZÁLEZ

- **Grado en Ingeniería Informática**
  - Universidad de Alicante
  - 2011 – 2015
- **Máster en Informática Gráfica, Videojuegos y RV**
  - Universidad Rey Juan Carlos
  - 2015 – 2017
- **Doctorado en Informática (Deep Learning and Computer Vision)**
  - Univerisdad de Alicante
  - 2018 – Actualidad

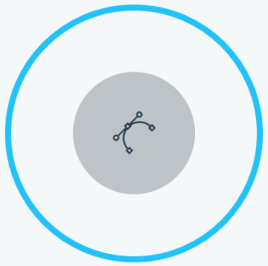
MAIL: **PMARTINEZ @ DTIC.UA.ES**

DEDICACIÓN

INVESTIGACIÓN

3DPLAB

The 3D Perception Lab at the University of Alicante is a group of researchers interested in the intersection of machine learning and computer vision. Our research mission focuses on various aspects of perception often related with mobile robotics in which we exploit 3D data as the main source of information. Some of our research lines include object recognition, semantic segmentation, rigid and non-rigid registration, visual localization and mapping, behavior analysis, and depth estimation. Apart from those general lines we are also highly interested in making those solutions run efficiently by leveraging GPU acceleration using CUDA. Aside from 3D data as our backbone, we are also tied together by our shared vision in the great potential of artificial intelligence, mainly deep learning, which we try to apply and push its limits in every project we work on.



DEEP LEARNING

Artificial intelligence applied to computer vision and robotics (semantic segmentation, depth estimation, scene understanding, object recognition, localization and maping).

3D COMPUTER VISION

Traditional computer vision methods and new challenges for 3D data (rigid registration, non-rigid registration, reconstruction).

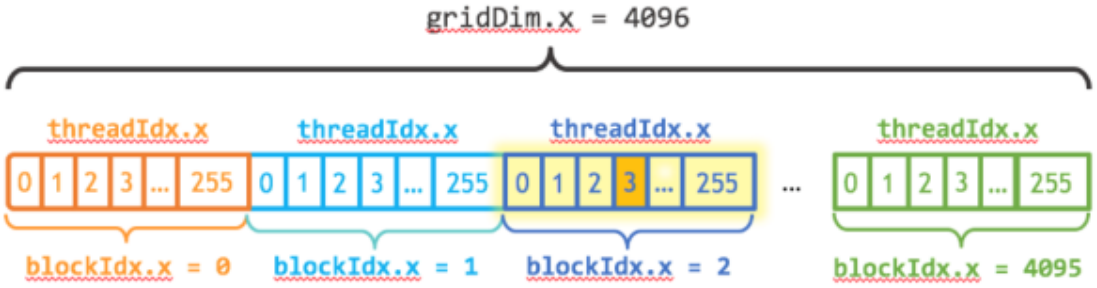
GPU COMPUTING

Acceleration of computer vision methods and artificial intelligence pipelines for real-time execution and maximum efficiency.

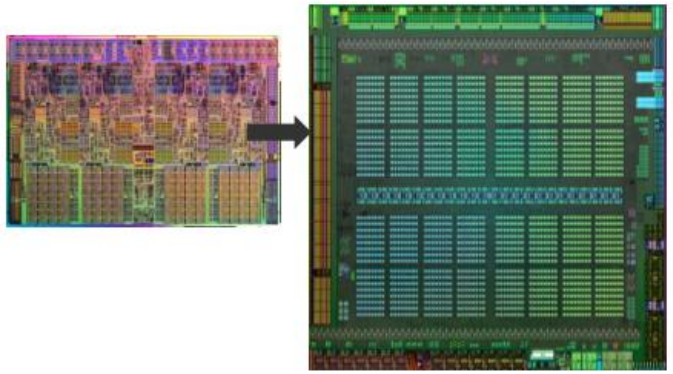
ESTRUCTURA

- TEORÍA:
  - HISTORIA DE LA GPGPU
  - ARQUITECTURA CUDA

- PRÁCTICAS:
  - SUMA DE VECTORES




```
index = blockIdx.x  * blockDim.x + threadIdx.x
index =    (2)      *    (256)    +     (3)      = 515
```



CONTENIDO

LA LEY DE MOORE

GORDON MOORE

LA LEY DE MOORE

GORDON MOORE

Cramming More Components onto Integrated Circuits. Gordon E. Moore, 1965

“ THE NUMBER OF TRANSISTORS ON A CHIP DOUBLES EVERY 12 MONTHS”

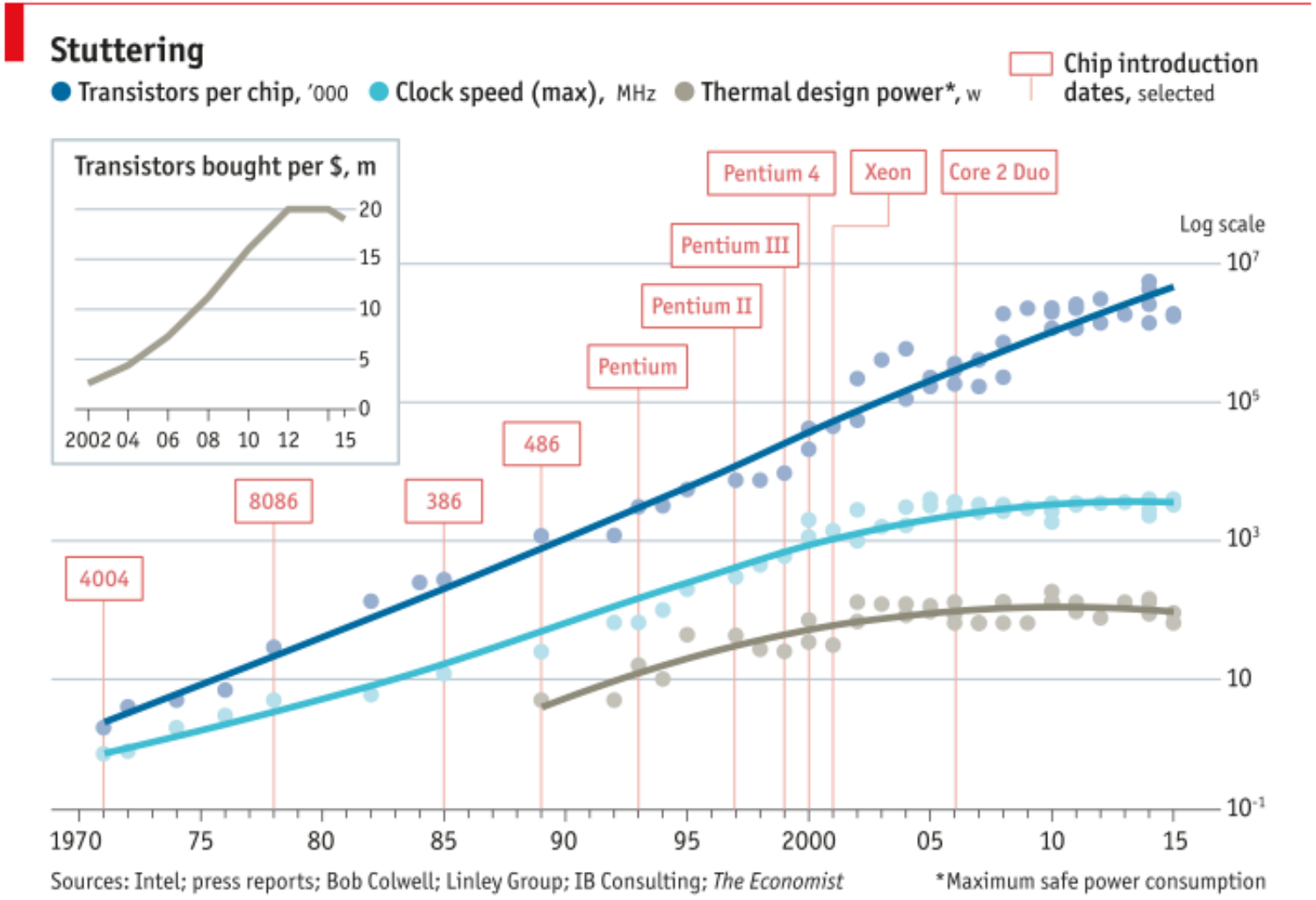
– GORDON MOORE, COFUNDADOR DE INTEL, 1965

“ THE NUMBER OF TRANSISTORS ON A CHIP DOUBLES EVERY 24 MONTHS”

– GORDON MOORE, COFUNDADOR DE INTEL, 1975

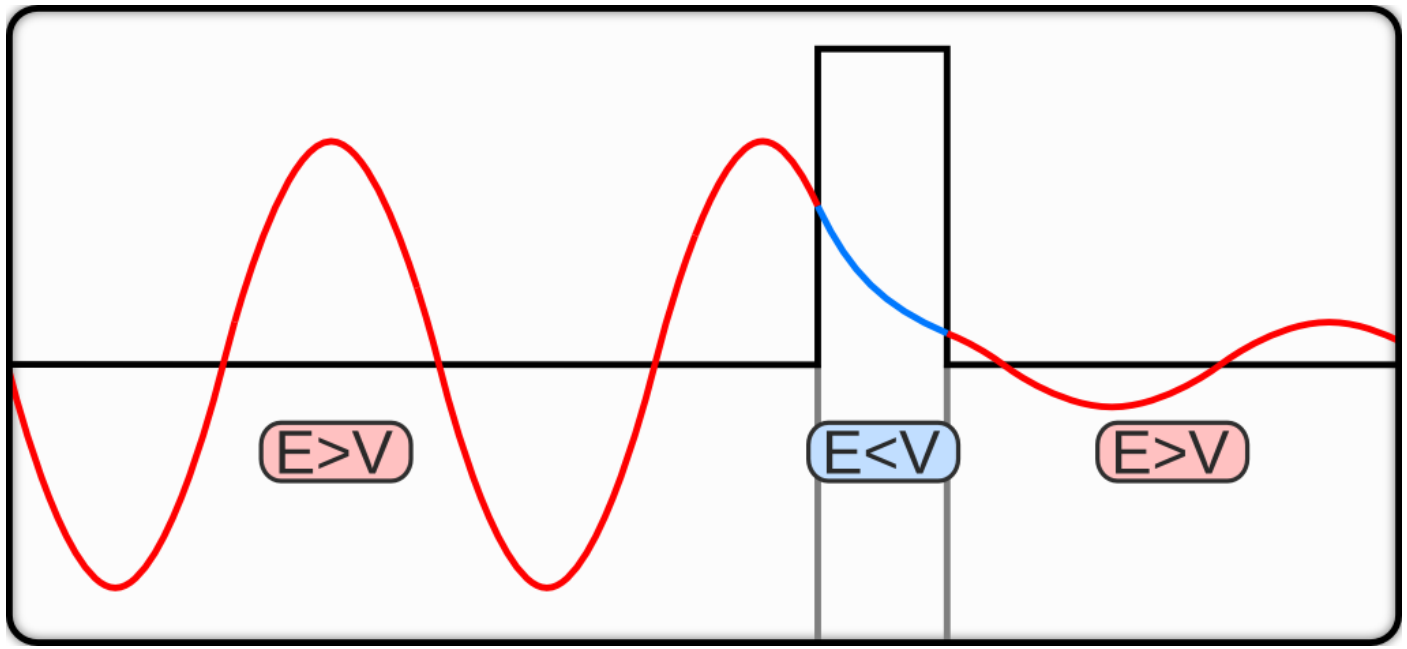
LA LEY DE MOORE

CADA DOS AÑOS, APROXIMADAMENTE, SE DUPLICA EL NÚMERO DE TRANSISTORES



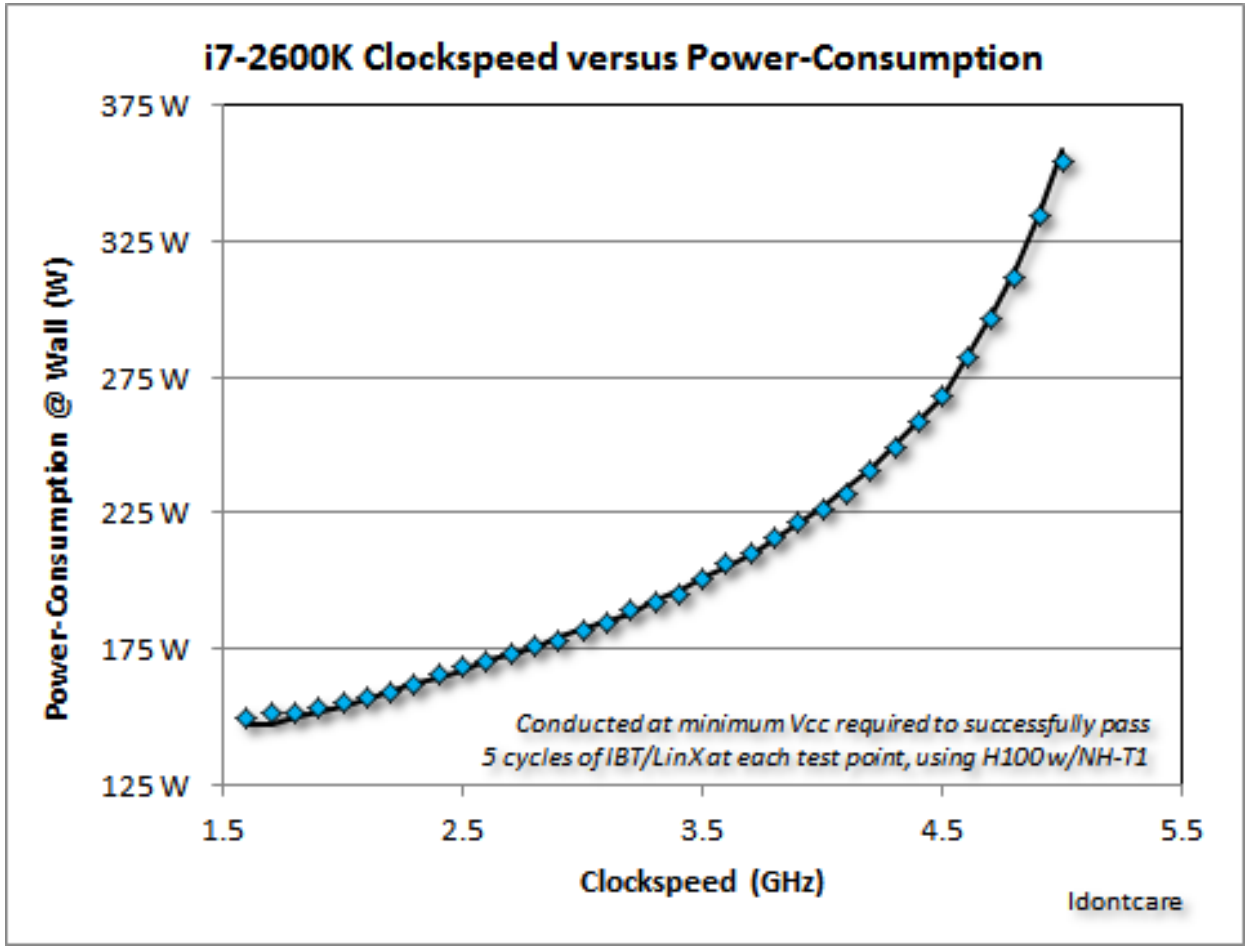
LA LEY DE MOORE

PROBLEMAS: TAMAÑO DEL TRANSISTOR Y EFECTO TÚNEL



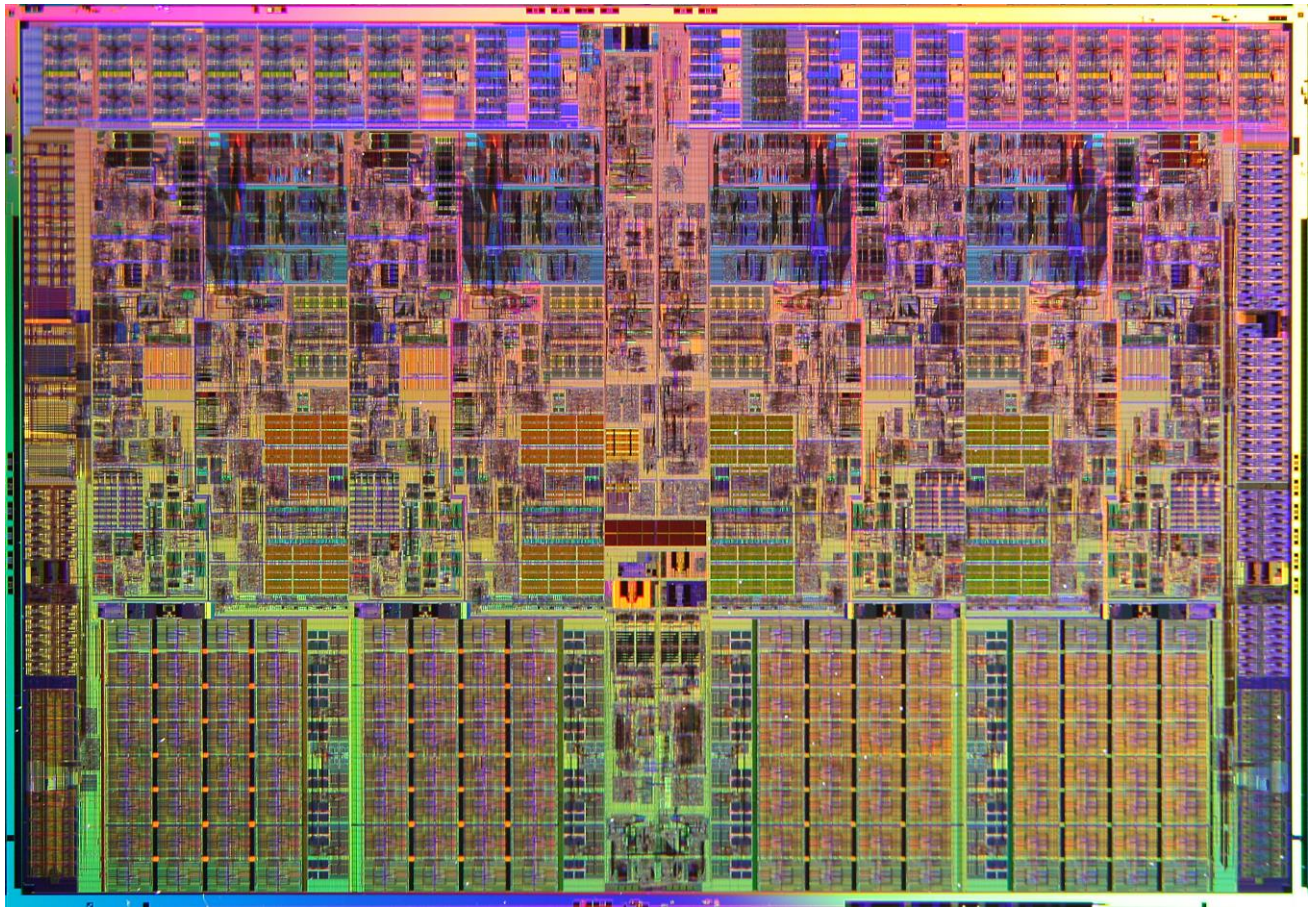
LA LEY DE MOORE

PROBLEMAS: CONSUMO ENERGÉTICO Y DISIPACIÓN DE CALOR



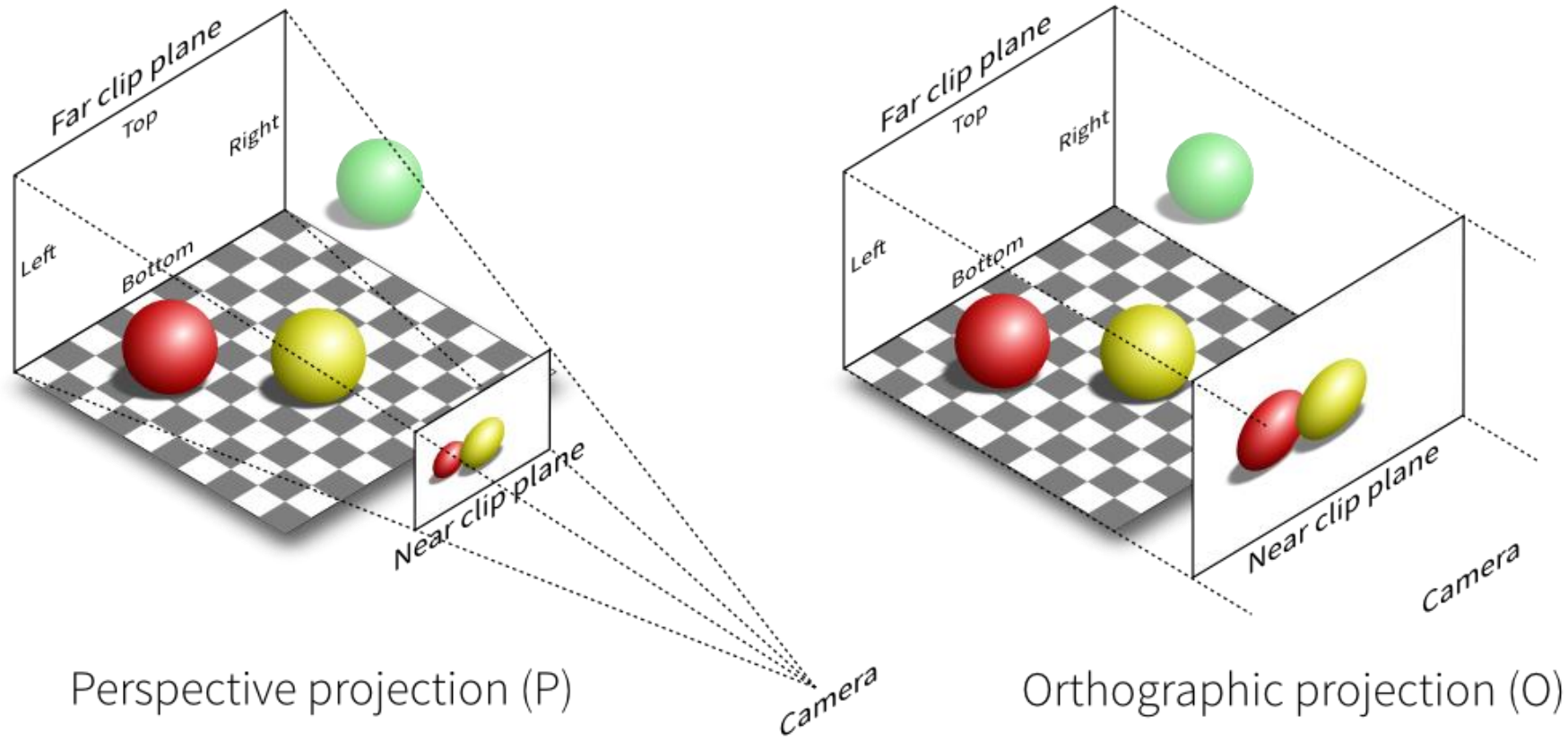
LA LEY DE MOORE

¿CÓMO CONTINUAR ESCALANDO? MULTICORE



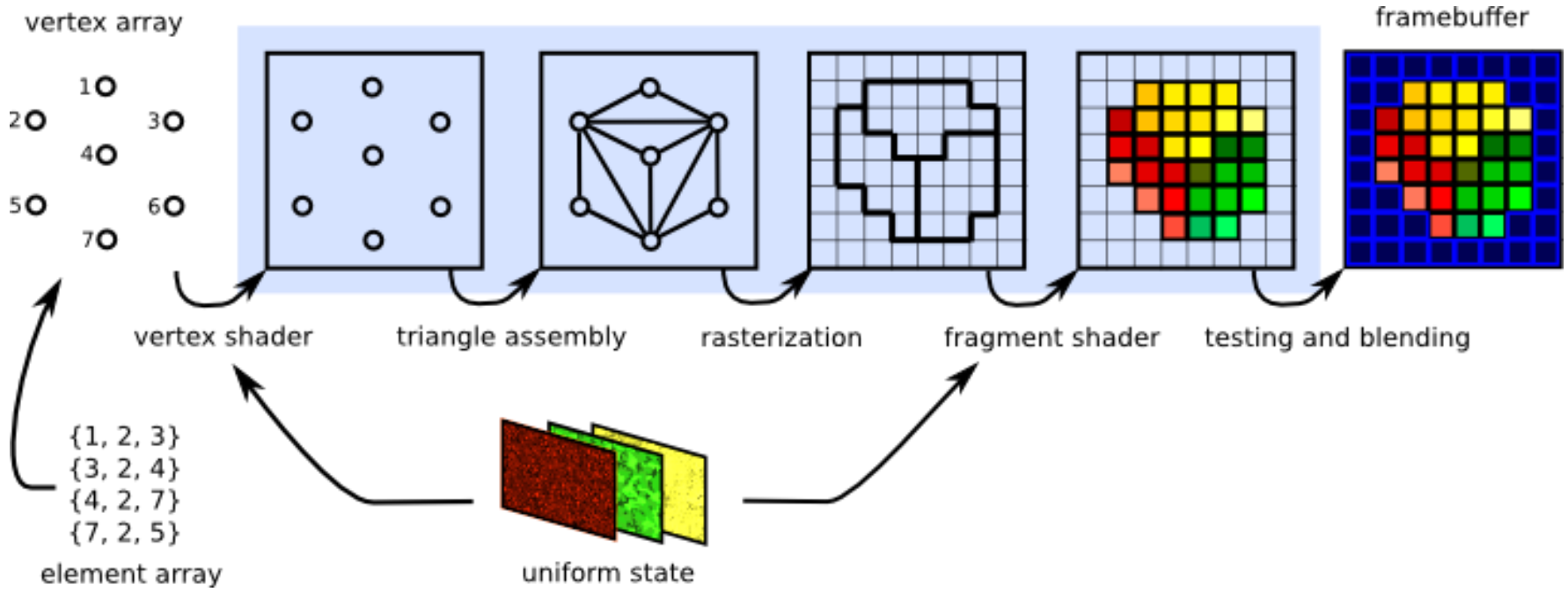
EL PIPELINE GRÁFICO

RENDERING



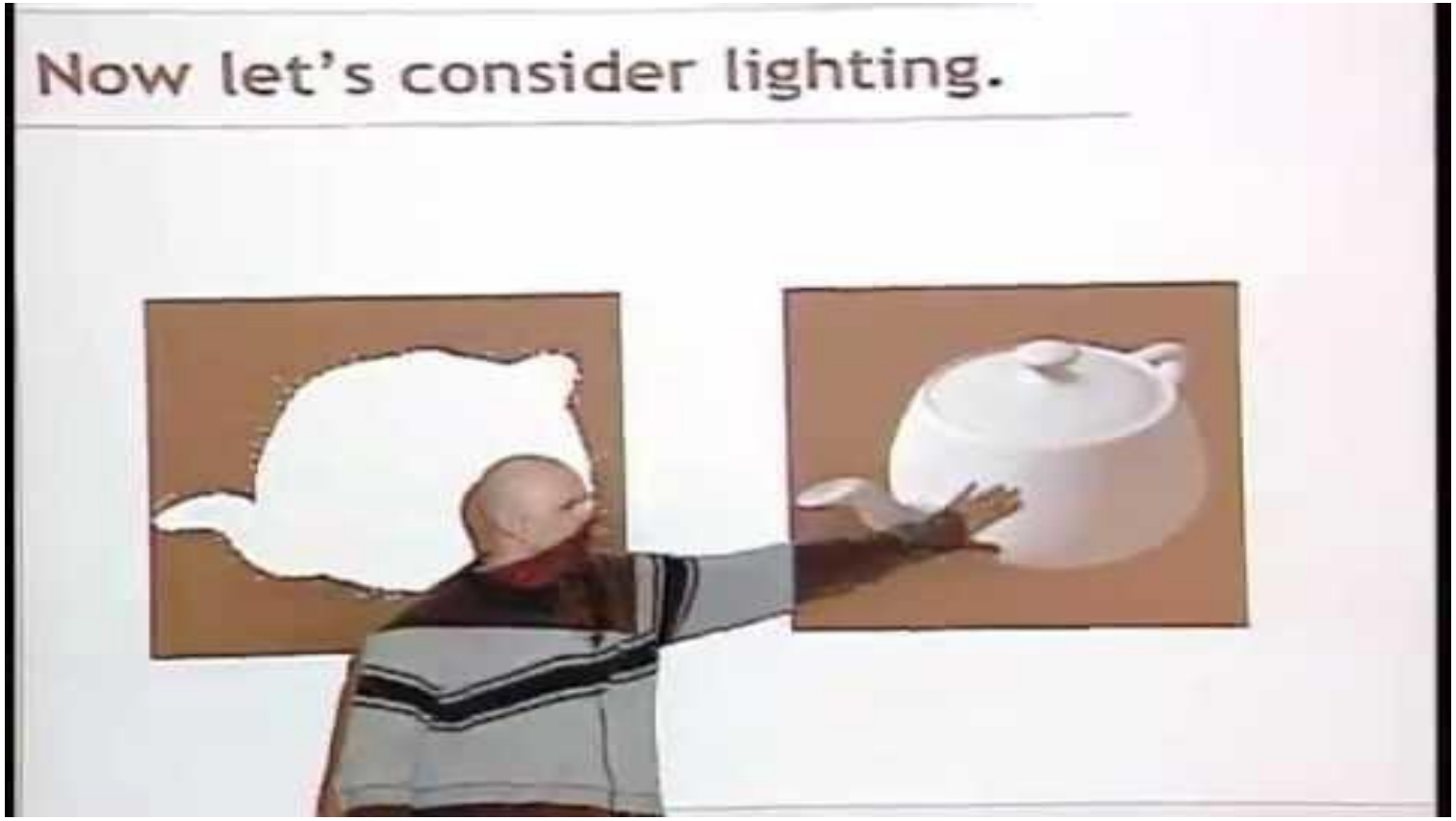
EL PIPELINE GRÁFICO

RENDERING



EL PIPELINE GRÁFICO

OpenGL GRAPHICS PIPELINE OVERVIEW por Owens



EL PIPELINE GRÁFICO

RAY TRACING

LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

GEFORCE 256 (1999)

LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

GEFORCE 256 (1999)

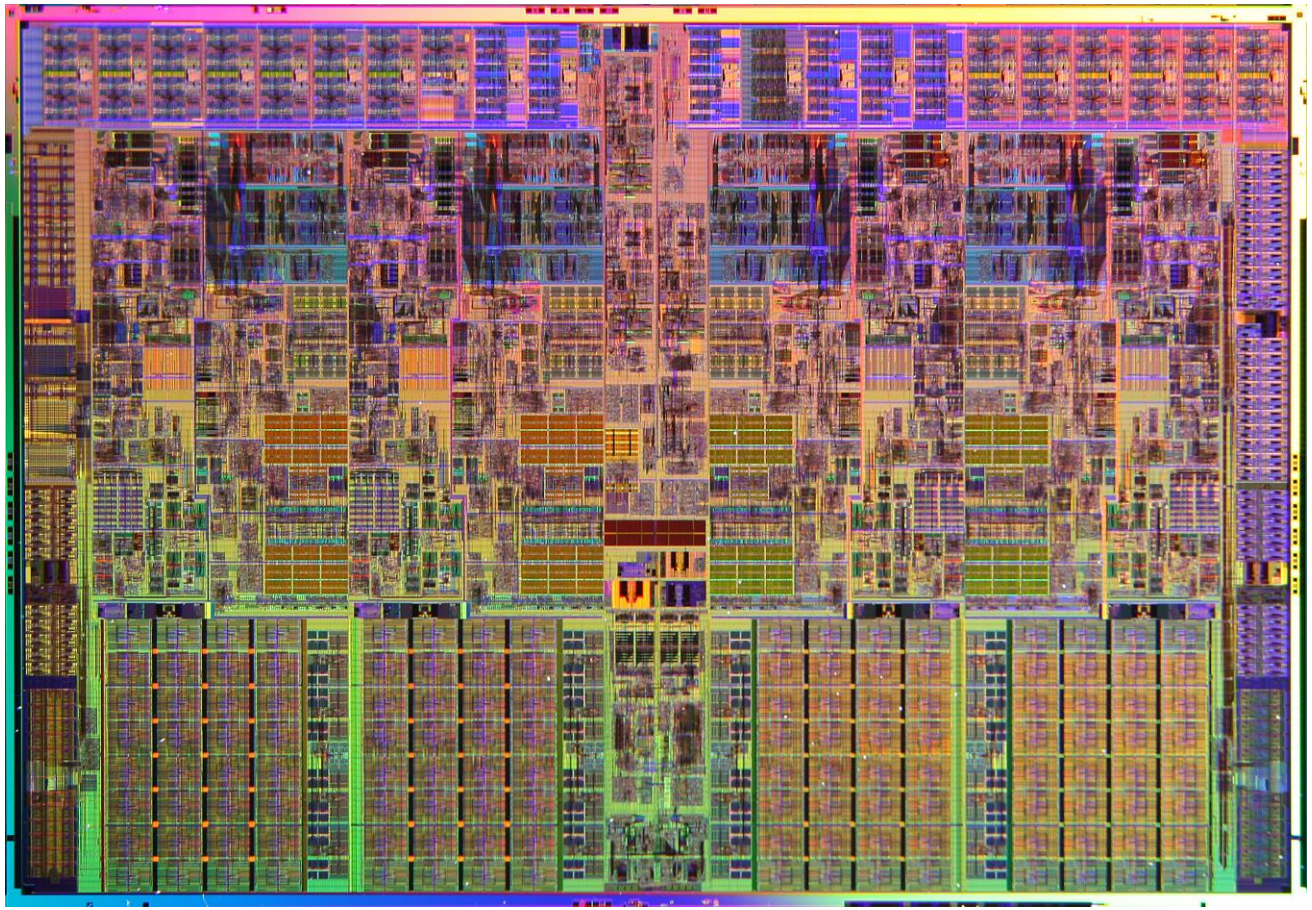
"A SINGLE-CHIP PROCESSOR WITH INTEGRATED TRANSFORM, LIGHTING, TRIANGLE SETUP/CLIPPING, AND RENDERING ENGINES THAT IS CAPABLE OF PROCESSING A MINIMUM OF 10 MILLION POLYGONS PER SECOND."

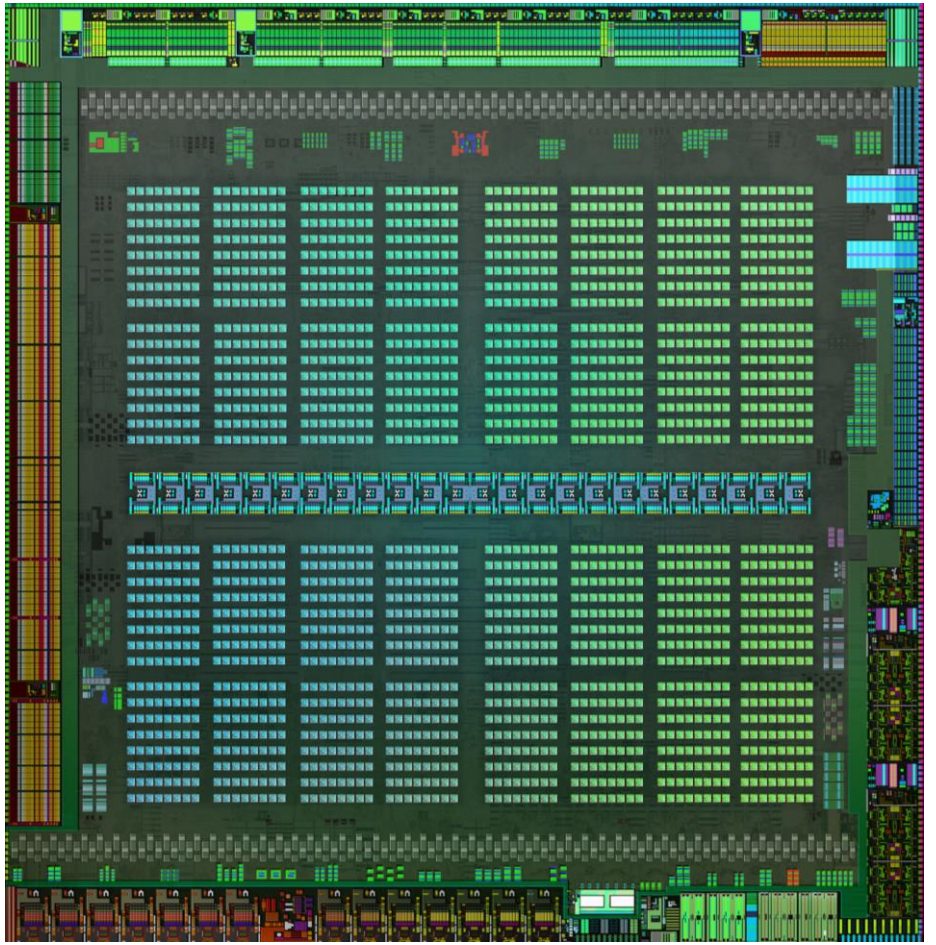
LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

GEFORCE 256 TECH DEMO

LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

LA ESENCIA DE LA GPU



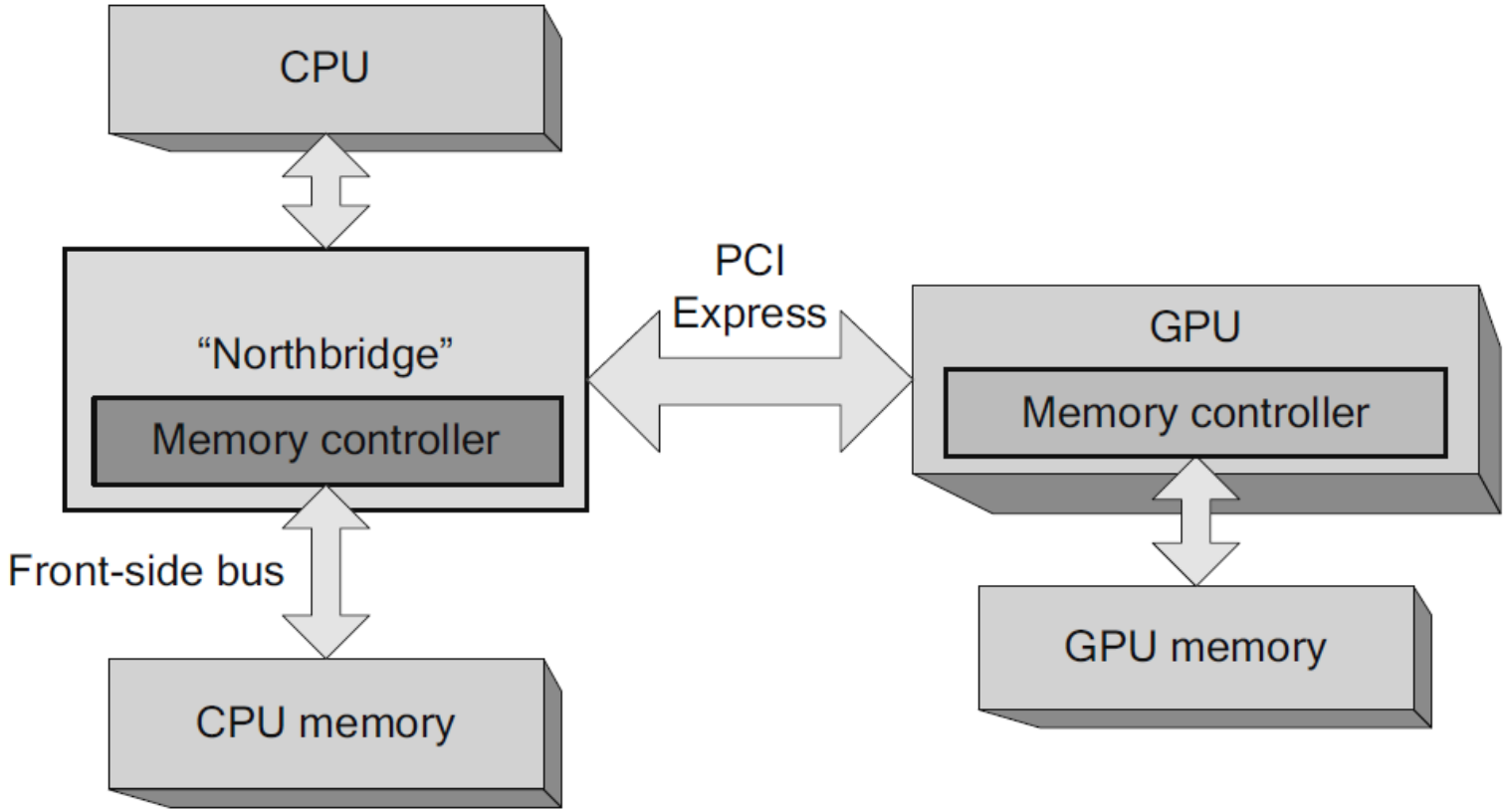


LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

DEJEMOS QUE LOS CAZADORES DE MITOS LO EXPLIQUEN

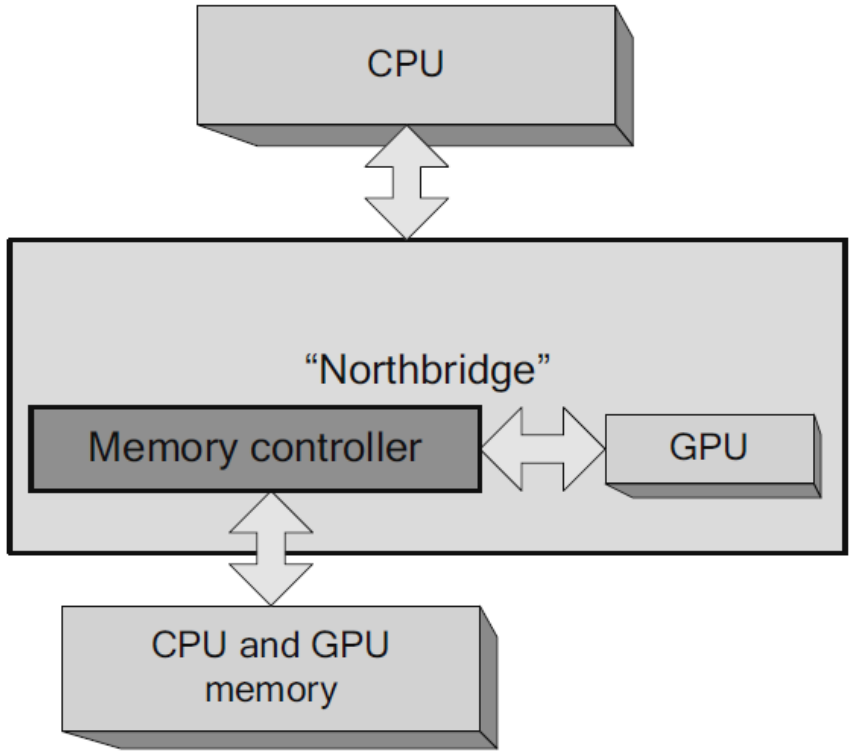
LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

ARQUITECTURA CPU/GPU TÍPICA



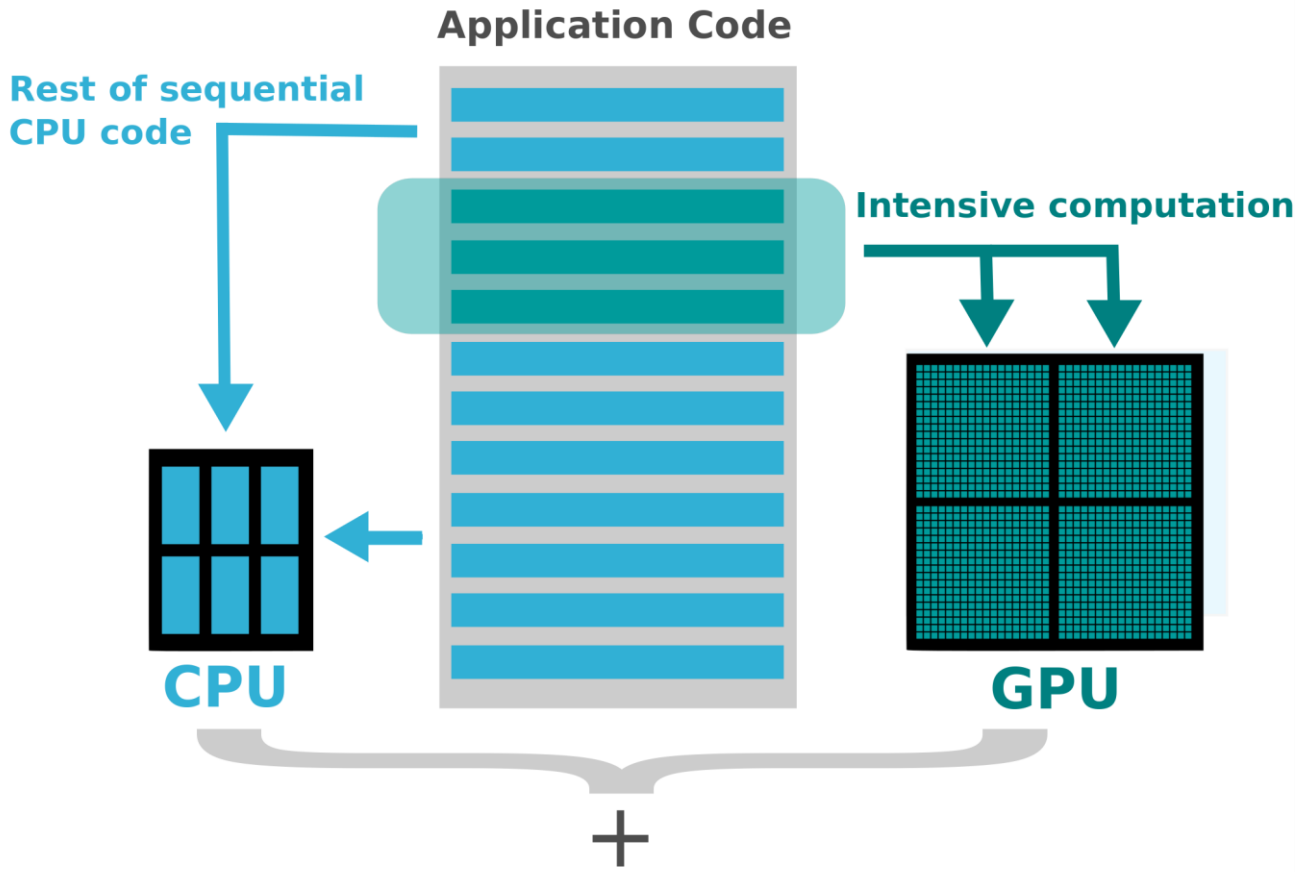
LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

ARQUITECTURA CPU/GPU INTEGRADA



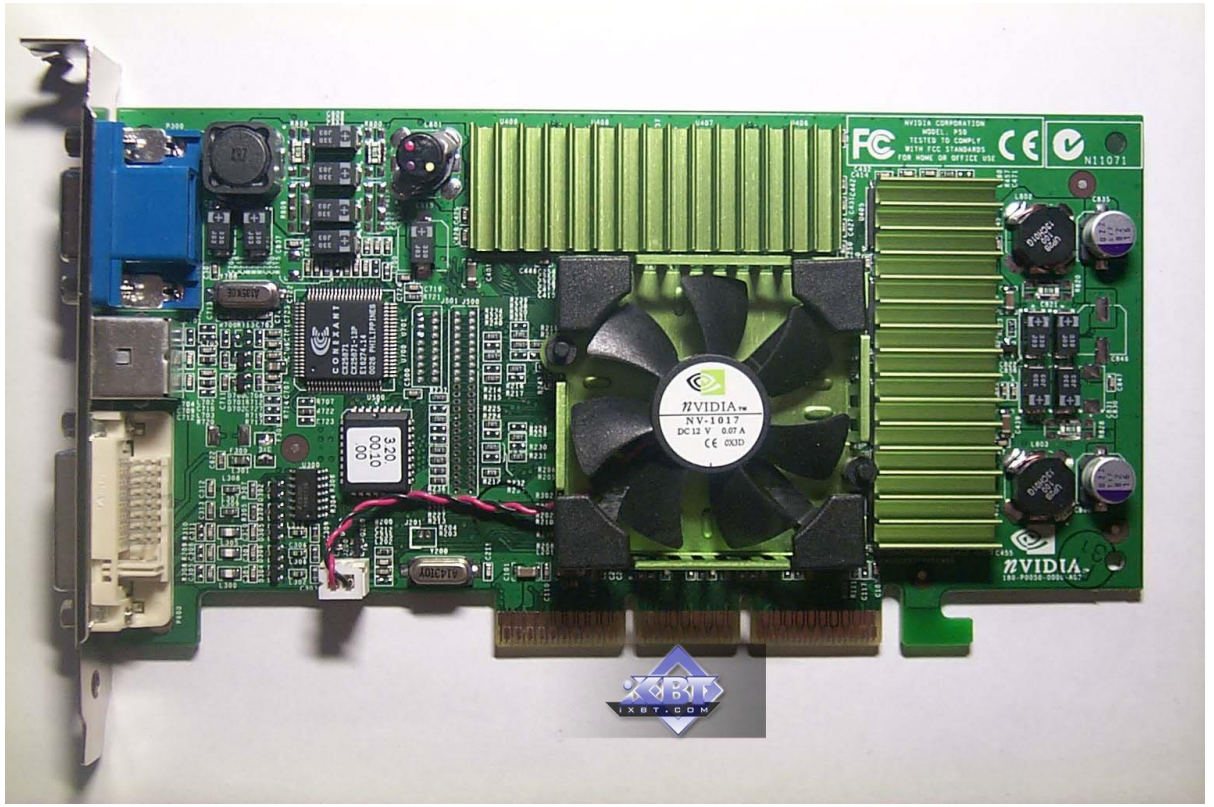
LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

COMPUTACIÓN HETEROGÉNEA



PRIMEROS PASOS EN COMPUTACIÓN SOBRE GPUs

GEFORCE 3 CON VERTEX Y PIXEL SHADERS PROGRAMABLES (2001)



PRIMEROS PASOS EN COMPUTACIÓN SOBRE GPUs

GEFORCE 3 TECH DEMO

PRIMEROS PASOS EN COMPUTACIÓN SOBRE GPUS

ENGAÑAR A LA GPU EMPLEANDO APIs GRÁFICAS (OPENGL O DIRECTX)

```
float saxpy (
        float2 coords : TEXCOORD0,
    uniform sampler2D textureY,
    uniform sampler2D textureX,
    uniform float alpha ) : COLOR
{
        float result;
        float yval=y_old[i];
        float y = tex2D(textureY,coords);
        float xval=x[i];
        float x = tex2D(textureX,coords);
        y_new[i]=yval+alpha*xval;
        result = y + alpha * x;
        return result;
}
```

Y = Y + alpha * X

PRIMEROS PASOS EN COMPUTACIÓN SOBRE GPUS

LIMITACIONES QUE IMPIDIERON EL PROGRESO

CURVA DE APRENDIZAJE DE OPENGL/DIRECTX Y ESFUERZO DE TRADUCCIÓN

NECESIDAD DE APRENDER LENGUAJES DE SHADING (CG, GLSL)

SOPORTE DE FLOAT O DOUBLE NO GARANTIZADO

LIMITACIONES EN LOS PATRONES DE ESCRITURA Y LECTURA DE MEMORIA

CARENCIA DE HERRAMIENTAS DE DEPURACIÓN Y CONTROL DE ERRORES

RECURSOS LIMITADOS: MEMORIA, VELOCIDAD, FLEXIBILIDAD...

ARQUITECTURA CUDA

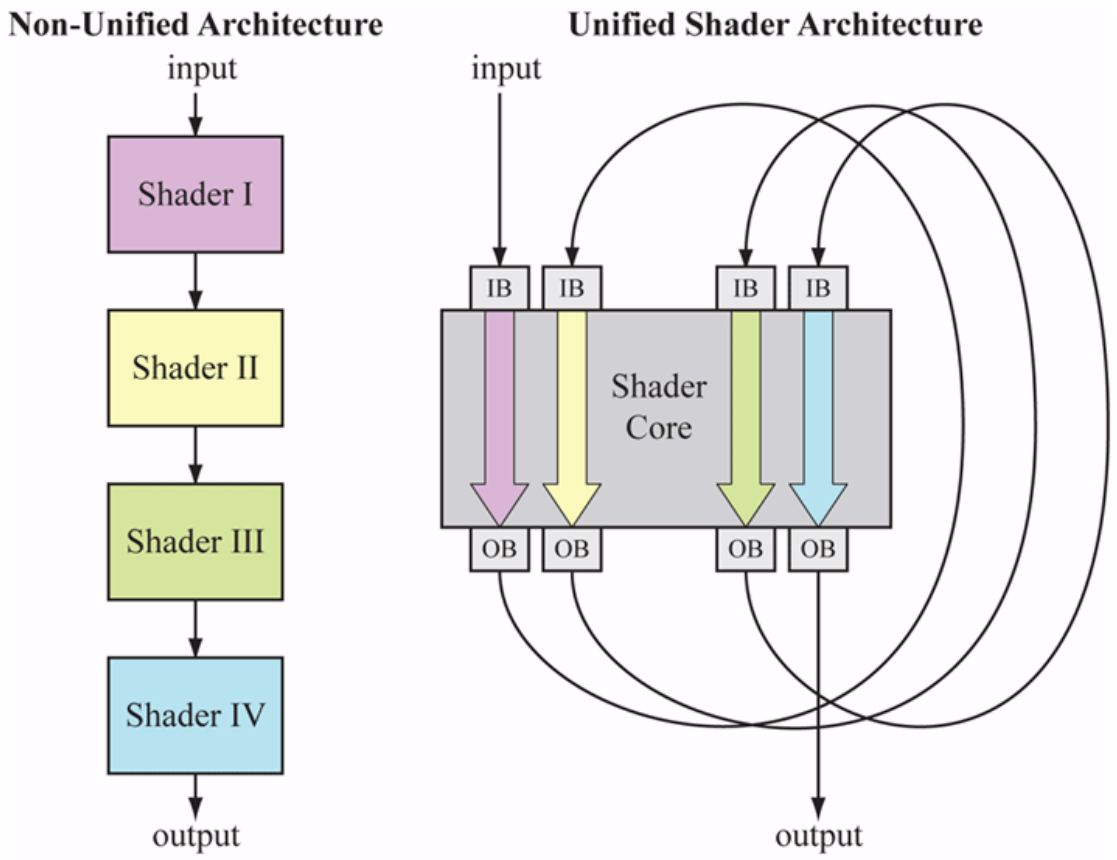
GEFORCE 8800 GTX (2007)

ARQUITECTURA CUDA

GEFORCE 8800 GTX TECH DEMO

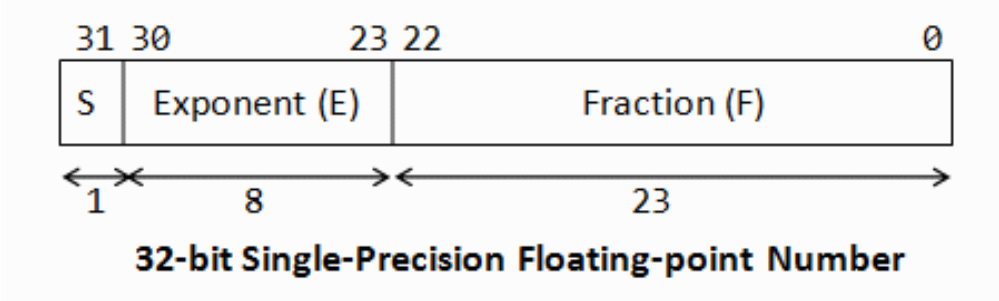
ARQUITECTURA CUDA

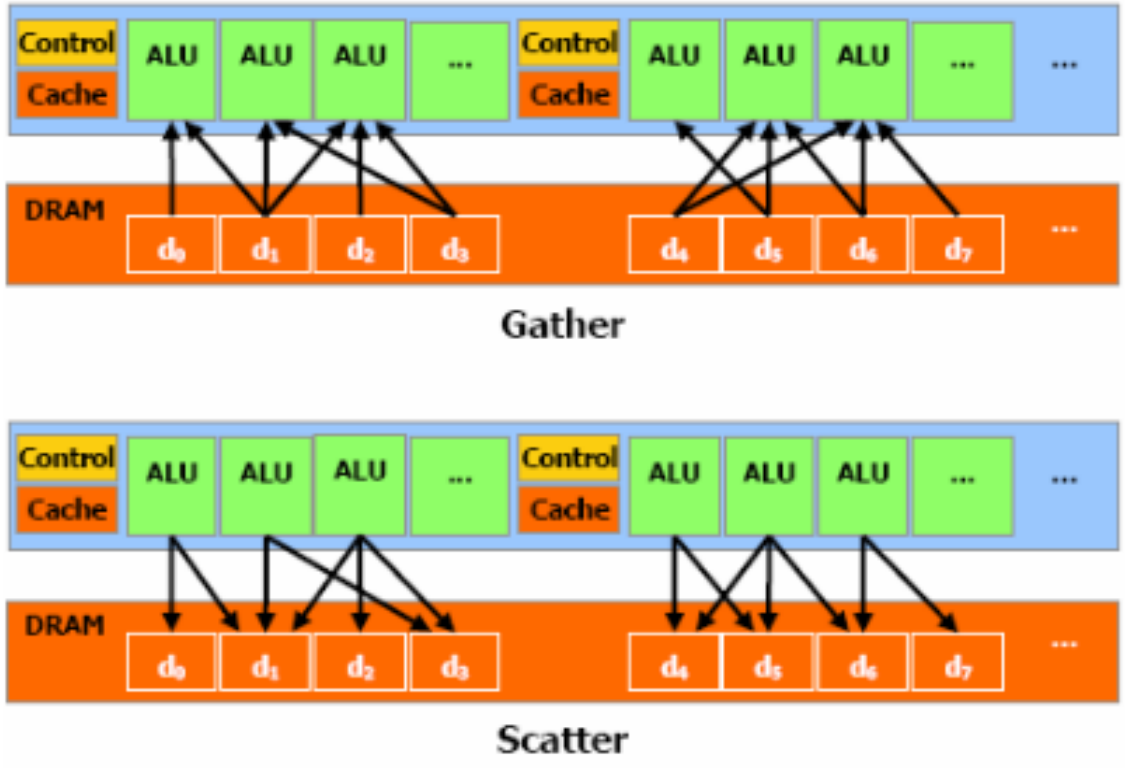
SHADERS UNIFICADOS



ARQUITECTURA CUDA

PRECISIÓN FLOAT Y PATRONES DE ACCESO





ARQUITECTURA CUDA

ECOSISTEMA

ARQUITECTURA HARDWARE PROPIA

DRIVER ESPECIALIZADO PARA LA GPU

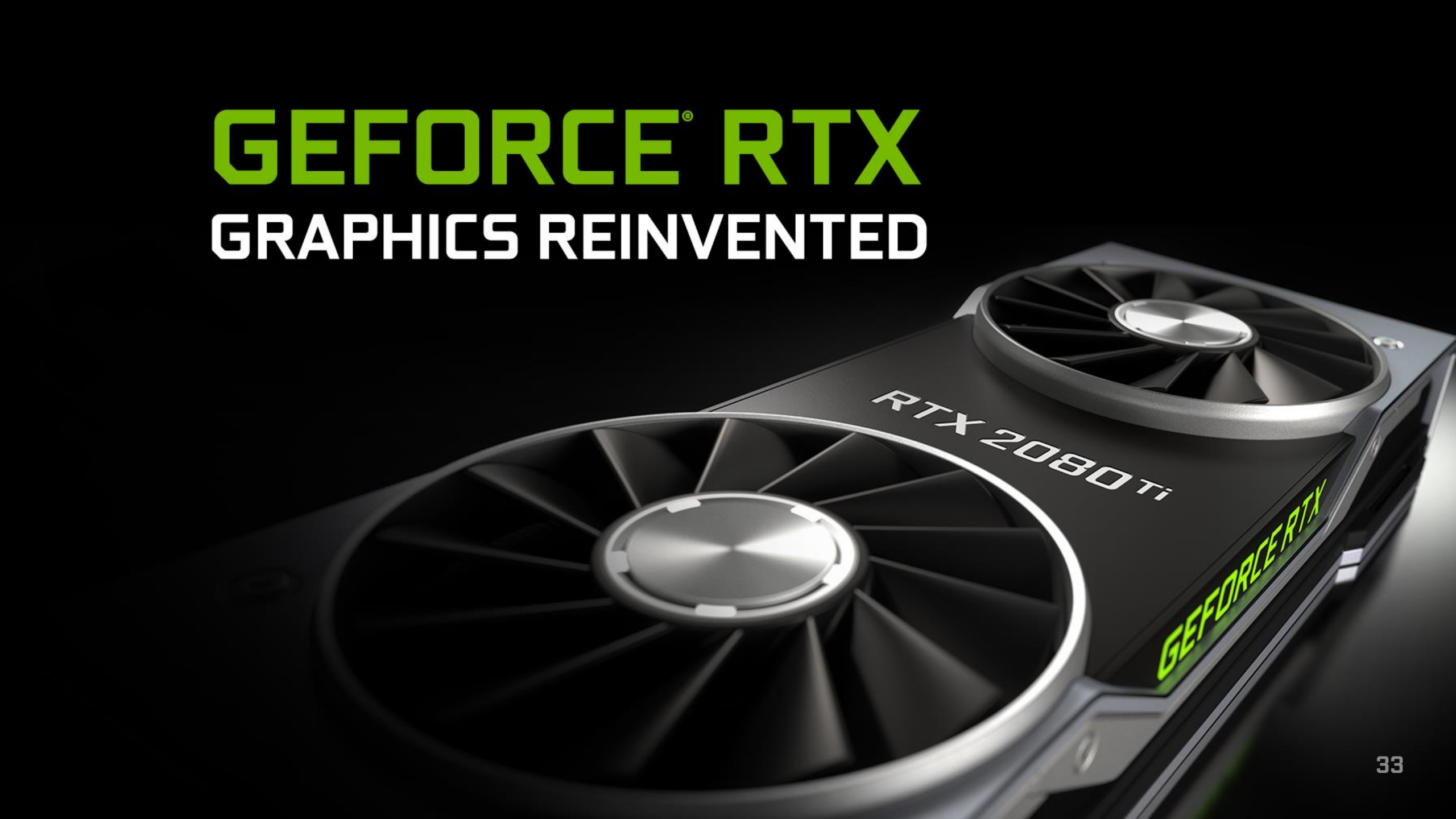
LENGUAJE DE PROGRAMACIÓN FLEXIBLE (BASADO EN C++ INICIALMENTE)

COMPILADOR Y ENTORNO DE DESARROLLO Y DEPURACIÓN

DOCUMENTACIÓN, TUTORIALES, DONACIONES

GEFORCE® RTX

GRAPHICS REINVENTED

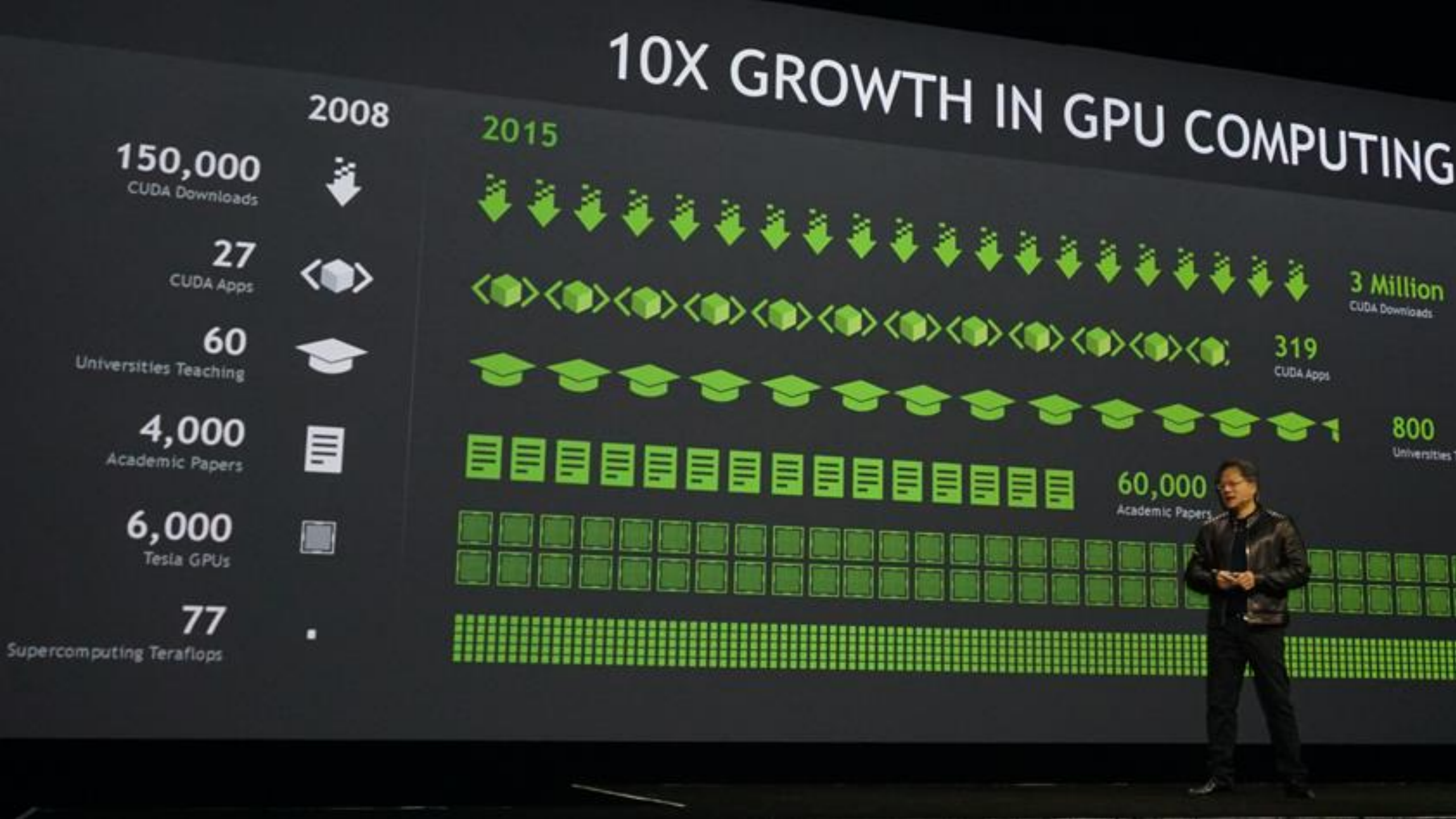


TURING

REAL-TIME RAYTRACING

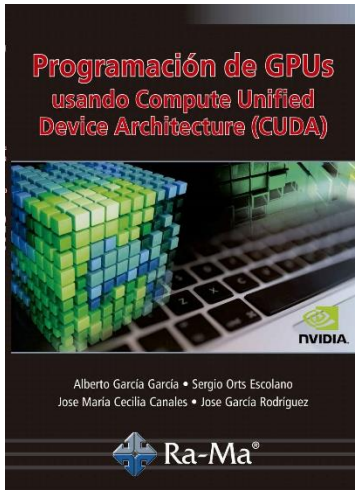
TURING

REAL-TIME SIMULATIONS


10X GROWTH IN GPU COMPUTING
2008
150,000
CUDA Downloads
27
CUDA Apps
60
Universities Teaching
4,000
Academic Papers
6,000
Tesla GPUs
77
Supercomputing Teraflops
2015
3 Million
CUDA Downloads
319
CUDA Apps
800
60,000
Academic Papers

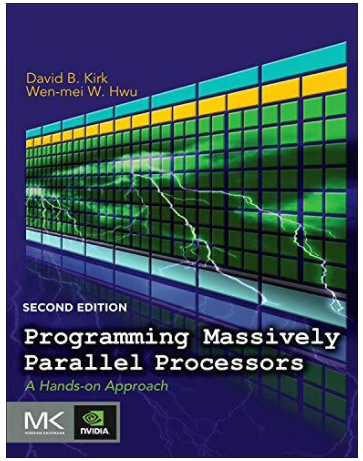
BIBLIOGRAFÍA RECOMENDADA

PARA APRENDER UN POQUITO MÁS



Programación de GPUs Usando Compute Unified Device Architecture (CUDA)

- Alberto García García, Sergio Orts Escolano, José Celilia Canales, José García Rodríguez

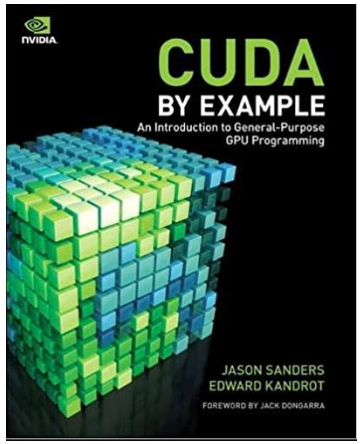


Programming Massively Parallel Processors (A Hands-on Approach)

- David B. Kirk, Wen-mei W. Hwu

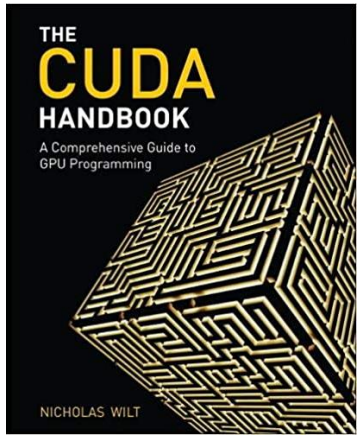
BIBLIOGRAFÍA RECOMENDADA

PARA APRENDER UN POQUITO MÁS



CUDA BY EXAMPLE (An Introduction to General-Purpose GPU Programming)

- Jason Sanders, Edward Kandrot



THE CUDA HANDBOOK (A Comprehensive Guide to GPU Programming)

- Nicholas Wilt

INICIOS Y EVOLUCIÓN DE LOS PROCESADORES GRÁFICOS (GPUs)

X JORNADAS DE DIVULGACIÓN DE APLICACIONES CIENTÍFICAS Y VISIÓN POR COMPUTADOR SOBRE PROCESADORES GRÁFICOS

UNIVERSIDAD DE ALICANTE

Pablo Martínez-González <pmartinez@dtic.ua.es>

Albert García-García <agarcia@dtic.ua.es>

José García-Rodríguez <jgarcia@dtic.ua.es>